Ademar Crotti Junior

UMA MÉTRICA *FUZZY* PARA APRENDIZAGEM DE ESTRUTURAS DE REDES BAYESIANAS PELO MÉTODO DE MONTE CARLO E CADEIAS DE MARKOV

Dissertação submetida ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Ciência da Computação.
Orientadora: Prof.ª Drª Silvia Modesto Nassar

Florianópolis
2014

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Crotti Junior, Ademar
Uma métrica fuzzy para aprendizagem de estruturas de redes bayesianas pelo método de Monte Carlo e Cadeias de Markov / Ademar Crotti Junior ; orientadora, Silvia Modesto Nassar - Florianópolis, SC, 2014.
72 p.

Dissertação (mestrado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Ciência da Computação.

Inclui referências

1. Ciência da Computação. 2. Redes Bayesianas. 3. Métrica fuzzy. 4. Monte Carlo e Cadeias de Markov. I. Modesto Nassar, Silvia. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Ciência da Computação. III. Título.

Ademar Crotti Junior

UMA MÉTRICA *FUZZY* PARA APRENDIZAGEM DE ESTRUTURAS DE REDES BAYESIANAS PELO MÉTODO DE MONTE CARLO E CADEIAS DE MARKOV

Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Ciência da Computação, e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Florianópolis, 19 de dezembro de 2014.

Prof. Ronaldo Mello, Dr.
Coordenador do Curso

Profª. Silvia Modesto Nassar, Drª.
Orientadora

Banca Examinadora:

Profª. Rosa Maria Viccari, Drª.
UFRGS

Prof. Paulo José de Freitas Filho, Dr.
UFSC

Prof. Mauro Roinsenberg, Dr.
UFSC

AGRADECIMENTOS

A minha orientadora, Silvia Modesto Nassar, pelo apoio e dedicação.

Aos professores Paulo José de Freitas Filho, Mauro Roinsenberg e Rosa Maria Viccari, por aceitarem o convite para a defesa final. Suas contribuições foram decisivas para a qualidade deste trabalho.

Aos familiares, pelos momentos com os quais não pude compartilhar.

Este trabalho é dedicado aos meus queridos pais.

RESUMO

A aprendizagem de estrutura de redes bayesianas (RB) a partir dos dados é considerada uma tarefa complexa, uma vez que o número de estruturas possíveis cresce exponencialmente de acordo com o número de variáveis. Existem dois métodos principais para esta tarefa de aprendizagem de estruturas de RB: o método de independência condicional, que busca uma estrutura consistente com os testes de independência realizados nos dados; o método de busca heurística, que explora o espaço de busca avaliando as possíveis estruturas por meio de algoritmos de busca. Além desses dois métodos, também são considerados os algoritmos híbridos, onde os dois métodos são aplicados na tarefa. A principal falha dessas abordagens tradicionais é que elas não conseguem identificar todas as relações existentes nos dados, sendo necessário investigar novas abordagem. Desta forma, esta pesquisa apresenta o desenvolvimento de uma métrica *fuzzy* de avaliação com um método de busca heurística para aprendizagem de estrutura de redes bayesianas, utilizando Monte Carlo via Cadeias de Markov. As diferentes métricas de avaliação de redes bayesianas utilizadas permitem identificar determinadas propriedades nas redes. Essas propriedades são determinadas em função da métrica aplicada. A combinação em uma métrica *fuzzy* possibilita avaliar diferentes propriedades simultaneamente. Os resultados deste trabalho foram avaliados no contexto de bases sintéticas por meio da comparação com outros algoritmos, convergência das cadeias de Markov e tempo de processamento. Os resultados evidenciam, apesar do tempo de processamento, que a métrica proposta, além de compatível com os algoritmos clássicos, melhorou o processo de avaliação de estruturas combinando diferentes métricas em uma métrica *fuzzy*.

Palavras-chave: Redes Bayesianas, Métrica *Fuzzy*, Monte Carlo e Cadeias de Markov.

ABSTRACT

Learning bayesian networks (BN) from data is considered a complex task, since the number of possible structures grows exponentially with the number of variables. There are two main approaches for learning BN: methods based on independence tests, seeking structures consistente with the tests performed on the data; methods based on heuristic search, exploring the search space with a search algorithm, evaluating the possible structures. Besides these two approaches, there are hybrid algorithms, where both methods are applied to the task. The main fault of these approaches is that they still fail to identify all existing relationships in the data, so it is necessary to investigate new approaches. This research presents the development of a fuzzy score metric in a heuristic search method for learning Bayesian network structures, in a Markov Chain Monte Carlo algorithm. Different score metrics used to learn BN structures identify certain properties in these networks. These properties are determined based on the score applied. The combination of these scores in a fuzzy metric enables the evaluation of different properties simultaneously. Results of this research were evaluated in the context of synthetic bases by comparing with other algorithms, convergence of Markov chains and processing time. The results show, despite the processing time, that the proposed metric is compatible with traditional algorithms, and improved the evaluation process of structures, combining different score metrics into a fuzzy metric.

Keywords: Bayesian networks, *Fuzzy* Metric, Markov Chain Monte Carlo.

LISTA DE FIGURAS

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AIC - *Akaike Information Criterion*
BDe - *Bayesian Dirichlet equivalence*
BN – Bayesian network
CS - *Collider Set*
DAG – *Directed Acyclic Graph*
EM-MCMC - *Expectation-Maximization* MCMC
GES – *Greedy Equivalence Search*
GS – *Grow-Shrink*
KGES – *k-Greedy Equivalence Search*
MCMC – Markov Chain Monte Carlo
MDL - *Minumum Description Length*
MMHC - *Max-Min Hill-Climbing*
RB – Redes bayesianas

SUMÁRIO

1 INTRODUÇÃO

Redes bayesianas (RB) são modelos gráficos que descrevem a distribuição de probabilidades em um determinado domínio de aplicação e podem ser utilizadas nas mais diversas áreas (PEARL, 1988).

Uma rede bayesiana é um modelo de raciocínio de incerteza representada por meio de sua estrutura e seus parâmetros. A estrutura expressa a relação entre variáveis e os parâmetros determinam as distribuições condicionais de cada variável (CHENG; GREINER, 2001). As variáveis e suas relações de independência condicional são expressadas por meio de um grafo acíclico dirigido (*Directed Acyclic Graph* – DAG*)* (PEARL, 1988).

A aprendizagem de redes bayesianas pode ser vista como uma combinação das aprendizagens de parâmetros e estrutura. A aprendizagem de parâmetros estima as probabilidades condicionais, sendo considerada simples quando a estrutura da rede está bem definida. A aprendizagem de estrutura estima a topologia da rede, determinando as conexões entre as variáveis (CASTILLO, 1997).

Na revisão da literatura foram encontrados dois métodos principais na aprendizagem de estrutura de redes bayesianas. O primeiro método faz testes de independência condicional nos dados e busca uma estrutura consistente com as independências observadas. O principal problema desta abordagem é o número exponencial de testes de dependência (YAN; CERCONE, 2010).

O segundo método define uma função que avalia quão bem as dependências de uma estrutura representam os dados, localizando a estrutura mais simples que aumenta o valor desta função. Algoritmos deste método exploram o espaço de busca, avaliando as estruturas por meio de funções de avaliação. Entre os métodos mais utilizados estão algoritmos gulosos, genéticos e Monte Carlo via Cadeias de Markov. Um problema desta abordagem é o espaço de busca, ou seja, todas as possíveis estruturas (YAN; CERCONE, 2010).

Alguns métodos combinam as duas abordagens e são chamados de algoritmos híbridos. A abordagem mais comum em métodos híbridos é utilizar testes de independência para restringir o espaço de busca e então aplicar um método de busca heurística (TSAMARDINOS; CONSTANTIN, 2006).

Estes métodos possuem problemas, pois ainda não são capazes de aprender todas as relações existentes em uma base de dados. Portanto, novos algoritmos com diferentes abordagens estão sendo propostos. Desta forma esta pesquisa tem a seguinte questão: como definir um método de aprendizagem de estrutura de redes bayes*ianas* que represente melhor os dados considerando as informações heterogêneas dos dados?

Uma maneira de melhorar os resultados de métodos que utilizam algoritmos de busca no processo de aprendizagem de redes bayesianas é modificar a forma de avaliar a estrutura.

Assim, este trabalho foca no desenvolvimento de uma métrica *fuzzy*, que combina diferentes métricas no processo de avaliação de um algoritmo para aprendizagem de estrutura de redes bayesianas, utilizando Monte Carlo via Cadeias de Markov como método de busca. A métrica *fuzzy* adiciona flexibilidade na avaliação das estruturas, com o objetivo de melhorar o processo de avaliação.

1.1 JUSTIFICATIVA E MOTIVAÇÃO

Redes bayesianas são modelos probabilísticos que representam o conhecimento sob incerteza por aleatoriedade. Possuem utilização em diversas áreas, como na predição de comportamentos, processamento de linguagem natural, em robótica na detecção de comportamento e orientação, entre outros.

A definição da estrutura e parâmetros de uma rede bayesiana é geralmente feita por um especialista. Entretanto, esse processo pode ser demorado, custoso e complexo, devido à quantidade de variáveis, dados incompletos e em virtude da dificuldade na manutenção da estrutura, inviabilizando o processo em muitos casos. Com base nisso, são necessárias pesquisas para automatizar o processo de construção de redes bayesiana.

O problema da aprendizagem de redes bayesianas é encontrar uma estrutura simples, porém densa de informações importantes. O problema é conhecido como NP-Difícil (CHICKERING, 1996). A aprendizagem de estruturas de redes bayesianas pode ser utilizada na descoberta de conhecimento em bases de dados ou para modelagem de problemas em um domínio de aplicação.

O presente trabalho se enquadra na área de Ciência da Computação, na linha de pesquisa de Inteligência Computacional, sendo que um dos objetivos desta linha de pesquisa é a aprendizagem de máquina. A aprendizagem de máquina desenvolve sistemas que podem aprender a partir de dados, a fim de tomar decisões com base em um conhecimento prévio, melhorando o desempenho em determinada tarefa.

A principal contribuição dessa pesquisa está centralizada em sua capacidade de potencializar o processo de aprendizagem de redes bayesianas por meio da utilização de uma métrica *fuzzy*. Assim, é possível avançar no estado da arte das métricas tradicionais de aprendizagem de RB.

1.2 OBJETIVO GERAL

O objetivo geral desta pesquisa é desenvolver uma métrica *fuzzy* que combine diferentes funções de avaliações na aprendizagem de estruturas de redes bayesianas utilizando o método de Monte Carlo via Cadeias de Markov.

1.2.1 Objetivos específicos

- Investigar técnicas utilizadas para a aprendizagem de estrutura de redes bayesianas;
- Identificar estratégias para melhorar a avaliação das estruturas de RB;
- Definir uma métrica que utilize as estratégias identificadas;
- Testar e avaliar a métrica.

1.3 ESTRUTURA E ORGANIZAÇÃO

Esta dissertação está dividida em seis capítulos. O segundo capítulo descreve a fundamentação teórica com os conceitos necessários e fundamentais desta pesquisa. No terceiro capítulo é apresentado o estado da arte sobre estruturas e métricas de aprendizagem de RB. O quarto capítulo apresenta os detalhes e a estruturação da métrica proposta, além dos procedimentos metodológicos. Os resultados são apresentados no quinto capítulo. Por fim, são apresentadas as conclusões e trabalhos futuros no sexto capítulo.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta os principais conceitos utilizados nesta pesquisa. A seção 2.1 apresenta formalmente o conceito de Redes Bayesianas.

Conceitos de lógica *fuzzy* são apresentados na seção 2.2. A seção 2.3 descreve o conceito de aprendizagem de máquina. Finalmente, a seção 2.4 apresenta os principais métodos de aprendizagem de Redes bayesianas a partir dos dados, assim como o método utilizado nesta pesquisa.

2.1 REDES BAYESIANAS

As Redes bayesianas são representações gráficas da distribuição de probabilidades, elas modelam a incerteza e são compostas por sua estrutura e seus parâmetros.

A estrutura de uma rede bayesiana apresenta as dependências condicionais entre suas variáveis. Esta é definida como um grafo acíclico dirigido, em que os nós representam as variáveis aleatórias (X_1, X_2, ..., X_n), e os arcos as dependências condicionais. Um arco direcionado X_i para X_j indica que X_i é pai de X_j e X_j é filho de X_i (SMITH, 2010).

Os parâmetros são as medidas das probabilidades condicionais, ou seja, quantificam o efeito que um conjunto de variáveis X_i tem sobre suas variáveis relacionadas X_j. A probabilidade conjunta é determinada por (PEARL, 1988):

$$P(X) = \prod_{i=0}^{n-1} P(X_i|\pi_i) \tag{1}$$

Onde:
X_i = variáveis aleatórias;
π_i = conjunto de variáveis pais de X_i;
$P(X_i| \pi_i)$ = probabilidade de X_i condicionada a π_i.

O conhecimento de uma rede bayesiana é extraído da probabilidade condicional de um conjunto de variáveis de consulta de acordo com as variáveis de evidência. Esta probabilidade é calculada utilizando o Teorema de *Bayes* e este processo é chamado de inferência

(NEAPOLITAN , 2004). Os eventos A e B são valores particulares de um conjunto de variáveis X.

$$P(B|A) = \frac{P(A|B).P(B)}{P(A)} \tag{2}$$

Onde:

P(A|B) = probabilidade de ocorrer o evento A condicionado que o evento B ocorreu;

P(B|A) = probabilidade de ocorrer o evento B condicionado que o evento A ocorreu;

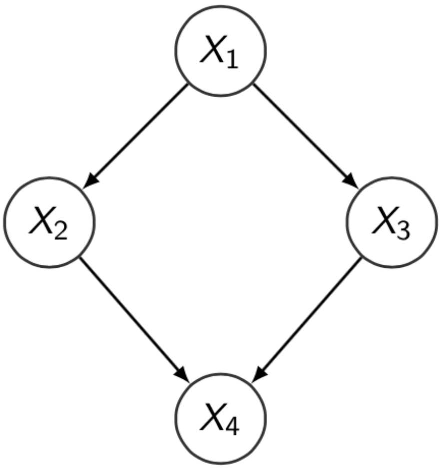
P(A) = Probabilidade do evento A ocorrer;

P(B) = Probabilidade do evento B ocorrer.

As redes bayesianas são consideradas ferramentas poderosas e são utilizadas como modelo de representação de conhecimento e inferência em condições de incerteza (PEARL, 1988).

A Figura 1 apresenta a estrutura de uma rede bayesiana. Quando uma variável X_1 possui relação com outra variável X_2, então existe um arco ligando estas variáveis.

Figura 1 – Estrutura de uma rede bayesiana.

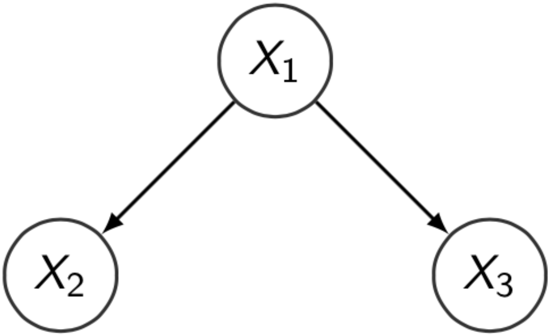


Fonte: Elaborada pelo autor

Quando a estrutura de uma rede bayesiana possui todos os nós independentes, dado o nó de saída, ela é chamada de Naïve Bayes ou classificador ingênuo (RUSSEL; NORVIG, 2009). A

Figura 2 apresenta um exemplo do classificador bayesiano Naïve Bayes.

Figura 2 – Estrutura do classificador bayesiano Naïve Bayes



Fonte: Elaborada pelo autor

A incerteza por aleatoriedade é modelada pela teoria de probabilidades. Diversos problemas possuem incerteza em função de conhecimento vago ou impreciso. A modelagem de incerteza por imprecisão é feita por meio da lógica *fuzzy*.

2.2 LÓGICA FUZZY

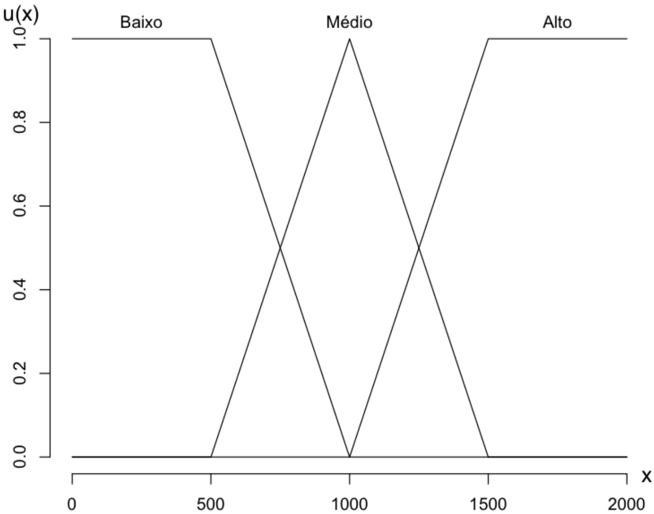
A teoria clássica de conjuntos associa um determinado elemento a um determinado grupo. Um conjunto *fuzzy* é representado por meio de um par ordenado, que define o elemento e o grau de pertinência deste elemento. A pertinência do elemento deve estar no intervalo [0,1].

A lógica *fuzzy* possibilita lidar com a imprecisão presente nos mais diversos problemas. Assim, um conjunto *fuzzy* define quanto um elemento satisfaz uma descrição vaga (RUSSEL; NORVIG, 2009).

Assim, a lógica *fuzzy* permite que os elementos pertençam a mais que um conjunto, por meio de graus de pertinência, que determinam quanto um elemento pertence a um determinado conjunto.

Os conjuntos *fuzzy* normalmente são identificados por meio de variáveis linguísticas. A Figura 3 apresenta funções de pertinência para uma determinada variável. Qualquer valor entre 500 e 1000 será considerado baixo e médio simultaneamente.

Figura 3 – Função de pertinência.



Fonte: Elaborada pelo autor

De acordo com Sivanandam; Sumathi e Deepa (2007), as funções de pertinência mais utilizadas são triangular, trapezoidal e gaussiana. Os detalhes destas funções são apresentados na sequência.

a) **função triangular**: tem forma de triangulo, e sua função geral é determinada por:

$$f(x;a,b,c) = \begin{cases} 0, x < a \\ \dfrac{x-a}{b-a}, a \leq x \leq b \\ \dfrac{c-x}{c-b}, b < x \leq c \\ 0, x > c \end{cases}$$

b) **função trapezoidal**: apresenta a forma de trapézio, onde sua função geral é definida por:

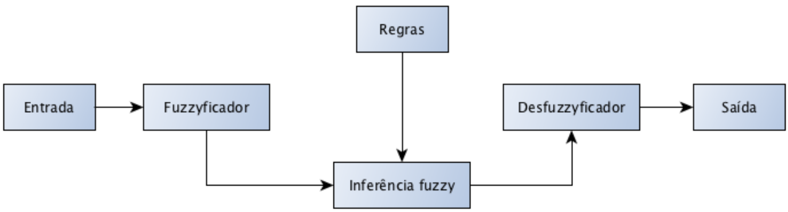
$$f(x;a,b,c,d) = \begin{cases} 0, x < a \\ \dfrac{x-a}{b-a}, a \le x \le b \\ 1, b < x \le c \\ \dfrac{d-x}{d-c}, c < x \le d \\ 0, x > d \end{cases}$$

c) **função gaussiana**: depende de dois parâmetros, o desvio padrão (σ) e a média (c):

$$f(x;\sigma,c) = e^{\frac{-(x-c)^2}{2\sigma^2}}$$

Além de funções de pertinência para os conjuntos *fuzzy*, um sistema *fuzzy* também precisa de uma base de regras, um motor de inferência e um método desfuzzyficador. A Figura 4 apresenta a estrutura de um sistema *fuzzy*.

Figura 4 – Sistema *fuzzy*.



O fuzzyficador converte valores reais em graus de pertinência a conjuntos *fuzzy*. As regras *fuzzy* consistem em um conjunto do tipo SE...ENTÃO onde variáveis de entrada são relacionadas a variáveis de saída.

A inferência em sistemas *fuzzy* é responsável por aplicar as regras e agregar as saídas. Entre os motores de inferência mais utilizados estão o Mamdani e o Larsen. O *Mamdani* utiliza o operador mínimo e o Larsen o produto para a implicação, ou seja, quando duas ou mais regras implicam sobre o mesmo conjunto *fuzzy*.

O processo de agregação combina os diferentes conjuntos *fuzzy* de saída, ativados pelas regras, em um único conjunto. O método mais utilizado para agregação é máximo, onde o conjunto com maior grau de pertinência é selecionado.

O passo final de um sistema *fuzzy* é a desfuzzyficação. Esta etapa converte os valores *fuzzy* encontrados pelo motor de inferência em valores reais.

Entre os principais para desfuzzyficação métodos destacam-se a média dos máximos, sendo composta pelo valor médio de todos os valores de saída. O método mínimo dos máximos, que seleciona o menor entre os valores de saída que tem grau de pertinência máximo e o máximo dos máximos que seleciona o maior valor. O centroide é o método mais utilizado sendo definido como (SIVANANDAM; SUMATHI; DEEPA, 2007):

$$C = \frac{\sum x_i \mu(x_i)}{\sum \mu(x_i)} \tag{3}$$

Onde:
x = valor da variável;
$\mu(x)$ = pertinência do valor *x* ao conjunto *A*

A lógica *fuzzy* permite modelar imprecisões com alta acurácia e é utilizada em diversas áreas. Em algoritmos de aprendizagem de máquina, a lógica *fuzzy* adiciona conhecimento e robustez ao processo de aprendizagem (HÜLLERMEIER, 2011).

2.3 APRENDIZAGEM DE MÁQUINA

Segundo Marsland (2009) aprendizagem de máquina pode ser definida como técnicas computacionais capazes de modificar ou adaptar suas ações, de modo que estas sejam aperfeiçoadas. O aperfeiçoamento é medido comparando as ações realizadas e as ações esperadas.

A aprendizagem de máquina também pode ser definida como o conjunto de métodos para detectar padrões, permitindo deste modo usá-los na predição de dados futuros, ou na tomada de decisão (MURPHY, 2012).

Em geral existem dois tipos de aprendizagem: aprendizagem supervisionada e não supervisionada. Na aprendizagem supervisionada, um conjunto de dados de treinamento com os resultados esperados é dado como entrada, e o algoritmo tenta generalizar para responder corretamente a qualquer outra entrada. Algoritmos de aprendizagem não supervisionada não recebem os resultados esperados e tentam identificar similaridades nos dados, para que estes sejam categorizados juntos (MARSLAND, 2009).

2.4 APRENDIZAGEM DE RB A PARTIR DOS DADOS

Redes bayesianas podem ser modeladas por um especialista ou aprendidas a partir dos dados. A modelagem feita por um especialista pode ser custosa, demorada e complexa, devido a dados incompletos, a quantidade de variáveis, entre outros fatores. Desta forma, o processo de aprendizagem a partir dos dados de forma automática tornou-se necessário.

O processo de aprendizagem pode ser dividido em duas etapas, a aprendizagem de estruturas e a aprendizagem de parâmetros (CASTILLO, 1997).

A aprendizagem de estrutura identifica as dependências entre as variáveis e a direção da causalidade. A aprendizagem de parâmetros ou das probabilidades, indica a intensidade dos relacionamentos da estrutura, sendo considerada uma tarefa simples quando a estrutura da rede é conhecida (CHENG; GREINER, 2001).

Existem duas abordagens principais na aprendizagem de estrutura de redes bayesianas.

2.4.1 Métodos com base em independência condicional

O método com base em independência condicional faz testes de independência condicional sobre os dados. As redes aprendidas tentam se aproximar das relações encontradas no conjunto de dados (LARRAÑAGA et al, 2013).

As métricas utilizadas neste método são baseadas na teoria da informação. A informação mútua de duas variáveis é dada por (CHENG et al., 2002):

$$I(X,Y) = \sum_{x,y} P(x,y) log \frac{P(x,y)}{P(x)P(y)} = \sum_{x,y} P(x,y) I(x,y) \qquad (4)$$

A fórmula da informação mútua condicional é dada por (CHENG et al., 2002):

$$I(X,Y|C) = \sum_{x,y} P(x,y) log \frac{P(x,y|c)}{P(x|c)P(y|c)} = \sum_{x,y} P(x,y) I(x,y|c) \qquad (5)$$

Onde:
X, Y = variáveis a serem testadas
C = subconjunto de variáveis condicionais
P = probabilidade observada no conjunto de dados

Quando I(X, Y| C) for menor que um determinado valor e, é dito que X é independente de Y dado C, quando C for vazio, X é dependente de Y. O valor e pode ser informado por um especialista ou calculado por algum método estatístico (YAN; CERCONE, 2010).

2.4.2 Métodos com base em busca heurística

Algoritmos que utilizam o método de busca heurística procuram pela estrutura que melhor represente os dados. O método aplica um mecanismo de busca, que com o auxílio de algum critério de qualidade avalia as estruturas candidatas (DALY; QIANG; STUART, 2011). Um exemplo de algoritmo que aplica este método é o K2.

Existem dois tipos de busca, *forward*, que inicia com um grafo totalmente desconectado, e *backward* que inicia com um grafo totalmente conectado (SCHWAIGHOFER, 2007).

Nas duas formas, a estrutura é avaliada e o processo é repetido até encontrar a melhor estrutura (YAN; CERCONE, 2010). De forma geral, esta estrutura deve satisfazer a uma função de pontuação, em que G é uma estrutura e D um conjunto de dados:

$$Score(G, D) = argmax_G P(G|D) \tag{6}$$

Onde:
G = a rede bayesiana;
D = o conjunto de dados.

Aplicando o teorema de Bayes tem-se a seguinte equação:

$$Score(G, D) = argmax_G \frac{P(D|G)P(G)}{P(D)} \tag{7}$$

Como o objetivo é maximizar esta função, é necessário apenas maximizar o numerador, já que o denominador não depende de G. Assim, aplica-se a seguinte equação:

$$Score(G, D) = argmax_G P(D|G)P(G) \quad (8)$$

Existem diversas funções de pontuação para avaliação de redes bayesianas, algumas favorecem estruturas mais complexas, outras penalizam a complexidade da rede. Além disso, uma outra importante característica dessas métricas é a capacidade de decomposição. Isto faz com que o cálculo seja a combinação de fatores menores, assim, pequenas alterações na estrutura podem ser facilmente computadas (LARRAÑAGA et al., 2013). O método de Monte Carlo via Cadeias de Markov explora o espaço de busca, utilizando uma métrica na avaliação das estruturas.

2.4.2.1 Monte Carlo via Cadeias de Markov

Monte Carlo via Cadeias de Markov (MCMC) é um método que utiliza aproximação por amostragem. O objetivo é gerar uma cadeia de Markov cujo limite seja a distribuição desejada (BROOKS, 2011).

O método obtém amostras aleatórias de uma distribuição de probabilidade considerada difícil de amostrar diretamente. Começando de qualquer ponto, à medida que o número de amostras aumenta, a cadeia se aproxima de sua distribuição de equilíbrio. Isto ocorre devido a ergodicidade, ou seja, todo estado deve ser acessível a partir de qualquer outro estado (BROOKS, 2011).

3.4.2.2 Simulação de Monte Carlo

Método ou simulação de Monte Carlo é baseado na geração de números aleatórios e probabilidade. O método tem sido muito utilizado para obter soluções aproximadas, em que é impossível ou não-viável obter uma solução determinística. Os seguintes conceitos são utilizados (FREITAS FILHO, 2008):

- Variáveis aleatórias: são variáveis cujo valores são randômicos.
- Funções densidade de probabilidade: representam a distribuição de probabilidade de uma variável aleatória.
- Geradores de números aleatórios: algoritmos que geram números obedecendo a uma certa aleatoriedade, simulando a aleatoriedade encontrada na natureza. Os números gerados são chamados de pseudoaleatórios,

normalmente definidos dentro do intervalo [0, 1], uniformemente distribuídos e sem correlação.

O método de Monte Carlo é um método estatístico onde se utiliza a geração de números aleatórios para a realização de uma simulação. Sendo assim, os números gerados devem refletir a probabilidade da ocorrência das variáveis aleatórias.

2.4.2.3 Cadeias de Markov

Uma cadeia é uma sequência de possíveis estados, em que a probabilidade de estar em um estado *s* em um tempo *t* é dado por uma função do estado anterior. Uma cadeia de Markov possui a propriedade de Markov, que diz que a probabilidade em um tempo *t* depende somente do estado em $t - 1$. Os estados são ligados por probabilidades de transição que representam a probabilidade de mudança de estado (BROOKS, 2011).

Dado uma cadeia, pode-se iniciar em um determinado estado e aleatoriamente mudar de estado de acordo com as probabilidades de transição. Quando o objetivo é explorar uma determinada distribuição de probabilidade, é preciso definir uma matriz de transição que reflita esta distribuição, chamada de distribuição estacionária dada pela equação (MARSLAND, 2009):

$$\pi(x') = \sum_x \pi(x) q(x \to x') \tag{9}$$

Onde:
x = estado anterior;
x' = estado atual;
π_i = distribuição alvo;
q = distribuição da proposta.

Isto também garante as propriedades de ergodicidade, onde qualquer estado pode ser acessado a partir de algum outro estado, e de reversibilidade, em que a probabilidade de mudança de um estado *s* para o estado *s'* é a mesma que estar em *s'* e ir para o estado *s* dada pela equação (RUSSEL; NORVIG, 2009):

$$\pi(x) q(x \to x') = \pi(x') q(x' \to x) \tag{10}$$

Onde:
x = estado anterior;
x' = estado atual;

π = distribuição alvo;
q = distribuição da proposta.

Todos os algoritmos MCMC são considerados casos especiais do algoritmo *Metropolis-Hastings*.

3.4.2.4 Algoritmo *Metropolis-Hastings*

O algoritmo teve sua primeira versão apresentada por Nicholas Metropolis em 1953. Em 1970 o método foi generalizado por W. Keith Hastings.

O algoritmo assume uma distribuição que se possa gerar amostras. A primeira amostra é chamada de estado inicial. Após gerar uma segunda amostra a probabilidade de aceitação é calculada utilizando (MARSLAND, 2009):

$$\mathrm{P}(\mathrm{x}, x') = \min\left(1, \frac{\pi(x')\mathrm{q}(\mathrm{x}|x')}{\pi(\mathrm{x})\mathrm{q}(x'|\mathrm{x})}\right) \tag{11}$$

Onde:
P(*x, x'*) = probabilidade de aceitação;
x = estado anterior;
x' = estado atual;
π = distribuição alvo;
q = distribuição da proposta.

A cada amostra aceita a cadeia se move em direção à distribuição alvo, como a cadeia é reversível o algoritmo explora estados proporcionais a sua distribuição (BROOKS, 2011).

O algoritmo pode levar muitas iterações para convergir, assim, os estados iniciais são geralmente descartados em uma etapa chamada de *burn-in*. A quantidade de estados a serem descartados depende da quantidade de iterações e da taxa de convergência da cadeia.

2.4.2.5 Algoritmo EM-MCMC

O algoritmo EM-MCMC foi baseado no algoritmo MCMC Metropolis-Hastings. Desta forma, a cada iteração, o algoritmo aleatoriamente modifica a estrutura da rede bayesiana.

Cada modificação gera uma nova amostra da cadeia de Markov. As possíveis modificações na estrutura são adição, inversão ou remoção de arcos. A seleção da operação é feita utilizando números aleatórios.

A aceitação da estrutura é feita utilizando a equação 17. Onde as probabilidades de aceitação são calculadas utilizando uma única métrica.

Diferentes abordagens foram propostas para o problema de aprender redes bayesianas sobre os dados. Os métodos MCMC se mostraram eficientes para esta tarefa, sendo este o escolhido nesta pesquisa.

Métodos de busca heurística utilizam uma métrica para avaliar as estruturas explorando este espaço de busca.

2.4.3 Métricas de avaliação de RB

Algumas das principais métricas utilizadas em algoritmos que aplicam o método de busca heurística são:

- **A métrica AIC** (*Akaike Information Criterion*) possui dois termos: um controlando entropia, baseada em entropia condicional e outro controlando complexidade. A entropia, em teoria da informação, é um valor não negativo que mede a incerteza e tende a zero quando o conhecimento é alto. A métrica AIC é dada por (AKAIKE, 1974):

$$AIC = H(G, D) + K \tag{12}$$

Sendo que:

$$H(G, D) = -N \sum_{i=1}^{n} \sum_{j=1}^{q_i} \sum_{k=1}^{r_i} \frac{N_{ijk}}{N} log \frac{N_{ijk}}{N_{ij}} \tag{13}$$

$$K = \sum_{i=1}^{n} (r_i - 1).\, q_i \tag{14}$$

Onde:
G = a rede bayesiana;
D = o conjunto de dados;
N = número de registros em D;
q_i = número de instanciações de π_i;
r_i = número de instanciações de x_i;

N_{ijk} = número de instâncias no conjunto de dados onde x_i possui o k-ésimo valor e os pais de x_i em π_i estão na j-ésima instância;

$N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

Este métrica combina conceitos de verossimilhança, verificando se os dados se ajustam a estrutura.

- **A métrica MDL** (*Minumum Description Length*) utiliza os mesmos termos da métrica AIC, com uma pequena diferença no segundo termo. A métrica MDL é dada por (BOUCKAERT, 1994):

$$MDL(G,D) = H(G,D) + \frac{K}{2} logN \tag{15}$$

Onde:
G = a rede bayesiana;
D = o conjunto de dados;
N = número de registros em D;
H = equação 13;

Esta métrica é conhecida por encontrar estruturas de redes bayesianas mais simples que a métrica AIC.

- **A métrica BDe** (*Bayesian Dirichlet equivalence*) maximiza a probabilidade da estrutura da rede de acordo com os dados, ou seja, a métrica utiliza a probabilidade condicional de cada variável da rede. A métrica BDe é dada por (HECKERMAN; GEIGER; CHICKERING, 1995):

$$BDe(G,D) = P(G) \prod_{i=0}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(r_i)}{\Gamma(r_i+N_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(r_i+N_{ijk})}{\Gamma(r_i)} \tag{16}$$

Onde:
G = a rede bayesiana;
D = o conjunto de dados;
P(G) = probabilidade a priori;
Γ = função gama;
r_i = número de instanciações de x_i;
N_{ijk} = número de instâncias no conjunto de dados onde x_i possui o k-ésimo valor e os pais de x_i em π_i estão na j-ésima instância;

$N_{ij} = \sum_{k=1}^{r_i} N_{ijk}$.

A função gama de um valor n é dada por:

$$\Gamma(n) = (n - 1)! \qquad (17)$$

Assim, a métrica AIC analisa a informação da rede, a MDL a complexidade e a BDe a probabilidade.

As métricas são aplicadas em algoritmos do método de busca heurística, individualmente, identificando estruturas compatíveis com as propriedades que estas métricas avaliam.

3 ESTADO DA ARTE

Redes bayesianas possuem grande aplicabilidade e por isso a aprendizagem automática a partir dos dados é uma área de pesquisa muito ativa.

A revisão da literatura realizada considerou artigos publicados em bibliotecas digitais. As bases de dados utilizadas foram: IEEE Xplore, ACM digital Library, Springer Link e ScienceDirect. Alguns artigos foram referenciados por publicações encontradas nessas bases dados, porém, foram encontrados em outros repositórios, como eventos, conferências onde foram publicados ou em ambientes de universidades.

A pesquisa considerou trabalhos publicados entre 1999 e 2014. Os termos de busca mais utilizados foram: *bayesian networks*, *learning bayesian networks from data*, *inducing bayesian networks* e *bayesian network structure learning*.

A seleção dos artigos foi realizada analisando métodos que obtiveram bons resultados e que foram relevantes para outros trabalhos. Ao total 20 artigos foram escolhidos, os detalhes e a síntese de cada um desses trabalhos são apresentados na sequência.

Margaritis e Thrun (1999) apresentaram o algoritmo Grow-Shrink (GS) que aprende a cobertura de Markov de cada variável para então definir uma rede que obedeça estas observações. A cobertura de Markov de uma determinada variável X é composta pelo subconjunto de variáveis que se observadas fazem com que a variável X fique condicionalmente independente de todas as outras variáveis. Outro algoritmo chamado Collider Set (CS) utiliza o mesmo método e foi apresentado por Pellet e Elisseeff (2008). O uso de coberturas de Markov também foi utilizado para diminuir o espaço de busca das estruturas nos trabalhos apresentados por Aliferis et al. (2010) e Bui e Jun (2012).

Um algoritmo genético com um sistema *fuzzy* para avaliação das estruturas foi proposto por Morales (2004). O algoritmo K2GA foi apresentado por Faulkner (2007) e utiliza um algoritmo genético para a busca da estrutura da rede. A avaliação dos indivíduos é feita utilizando a métrica do algoritmo K2. Um algoritmo para definir os possíveis pais de uma variável foi apresentado por Ko e Kim (2014), o resultado foi aplicado como entrada para o algoritmo K2.

A ordenação das variáveis informada como entrada para o algoritmo K2, reduz o número de possíveis estruturas. Todos as variáveis devem ser informadas na lista, indicando que apenas os atributos informados antes de x_i podem ser pais de x_i (COOPER;

HERSKOVITZ, 1992). O algoritmo é conhecido por ter baixa complexidade computacional e conseguir bons resultados quando a ordenação das variáveis é adequada (DALY; QIANG; STUART, 2011). Informar a ordem das variáveis requer conhecimento prévio sobre a base de dados, sendo que especialistas podem ter dificuldade em informar este parâmetro.

Chickering (2002) apresentou o algoritmo *Greedy Equivalence Search* (GES). Esse algoritmo utiliza um método guloso, assim a cada iteração o vizinho com melhor pontuação é selecionado. O algoritmo *k-Greedy Equivalence Search* (KGES), baseado no algoritmo GES, foi apresentado em 2002 por Nielsen, Kocka e Pena. Uma modificação para melhorar a performance computacional do algoritmo GES foi apresentado por Alonso-Barba et al. (2011).

Um método para restringir o espaço de busca e posteriormente a aplicação de um método de pontuação para aprender a estrutura foram apresentados por Gámez, Juan e Puerta (2011). Tsamardinos, Brown e Constantin (2006) apresentaram o algoritmo *Max-Min Hill-Climbing* (MMHC) que utiliza um método de independência condicional para restringir o espaço de busca para então aplicar um método de busca heurística. Yuan e Malone (2013) apresentaram duas heurísticas para restringir o espaço de busca. Zhang Y., Zhang W. e Xie (2013) restringiu o espaço de busca utilizando testes de independência e métricas estatísticas, na fase final do método proposto por Zhang Y., Zhang W. e Xie (2013) o algoritmo GES é aplicado.

Friedman e Koller (2003) apresentaram um método MCMC sobre o espaço da ordem das variáveis, que é menor que o das estruturas. A ordem encontrada é utilizada em uma segunda etapa que busca a estrutura utilizando outro método MCMC. Grzegorczyk e Husmeier (2008) utilizaram um método MCMC sobre o espaço das estruturas com uma alteração no método que gera a amostra da cadeia de Markov. Um outro método que utiliza o algoritmo *Expectation-Maximization* para aprender os parâmetros da rede enquanto um método MCMC busca as estruturas, o algoritmo foi chamado EM-MCMC (Guo; Li, 2009). Niinimaki, Parviainen e Koivisto (2012) apresentaram outro algoritmo que utiliza o método MCMC sobre a ordem das variáveis, mas ao invés de aplicar o método MCMC para gerar as amostras, utilizam um método *annealed importance*. Masegosa e Moral (2013) apresentaram dois algoritmos, um método de busca e outro MCMC, que utilizam como entrada a estrutura genérica da rede. Costa (2013) apresentou um novo método MCMC sobre o espaço de estruturas, em que alguns arcos não são alterados pela amostragem pois são considerados fortes, baseados em testes de independência condicional.

O Quadro 1 sintetiza o estado da arte dos métodos descritos neste capítulo de acordo com sua abordagem. Esses métodos apresentados no Quadro 1 são referentes a pesquisas entre 1999 a 2014.

Quadro 1 – Síntese dos métodos de aprendizagem de estrutura de redes

Abordagem	**Autores**
Algoritmo genético	Morales(2004)
	Faulkner (2007)
Algoritmo guloso	Chickering (2002)
	Nielsen, Kocka e Pena (2002)
	Alonso-Barba et al. (2011)
	Ko e Kim (2014)
Método híbrido	Tsamardinos e Constantin (2006)
	Gámez, Juan e Puerta (2011)
	Yuan e Malone (2013)
	Zhang Y., Zhang W. e Xie (2013)
Cobertura de Markov	Margaritis e Thrun (1999)
	Pellet e Elisseeff (2008)
	Aliferis et al. (2010)
	Bui e Jun (2012)
Método MCMC	Friedman e Koller (2003)
	Grzegorczyk e Husmeier (2008)
	Guo e Li (2009)
	Niinimaki, Parviainen e Koivisto (2012)
	Masegosa e Moral (2013)
	Costa (2013)

Fonte: Elaborado pelo autor

O Quadro 1 resume as principais abordagens aplicadas na aprendizagem de estrutura de redes bayesianas. Ao analisar o quadro percebe-se que esta área possui grandes esforços atualmente. As diferentes abordagens procuram melhorar a aprendizagem, representando melhor o conhecimento existente nos dados.

Os trabalhos analisados também mostram que o método MCMC está sendo utilizado em pesquisas recentes e de relevância na tarefa de aprendizagem de estrutura de redes bayesianas, sendo este o método utilizado nesta pesquisa.

4 MÉTRICA PROPOSTA

Este capítulo apresenta uma métrica *fuzzy* para avaliação de estruturas aplicado a um método para aprendizagem de estrutura de redes bayesianas utilizando Monte Carlo via Cadeias de Markov, chamado de MCMC-*Fuzzy*.

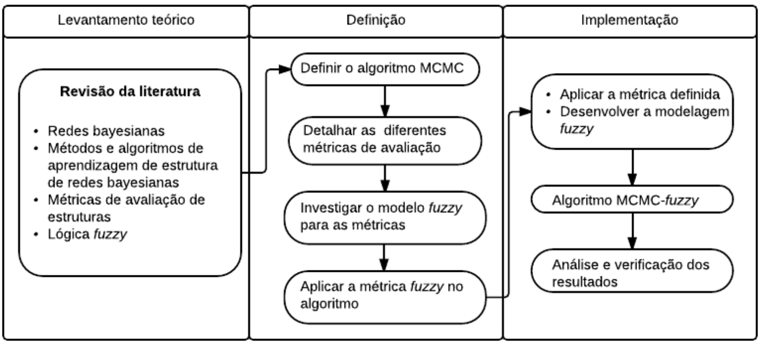
O método é baseado no algoritmo EM-MCMC e foi publicado nos anais do 25º Simpósio Brasileiro de Informática na Educação (SBIE) (CROTTI JUNIOR, 2014).

4.1 PROCEDIMENTOS METODOLOGICOS

O estudo realizado é de natureza aplicada e de base tecnológica, pois visa gerar conhecimentos dirigidos a solução de um problema específico através do empirismo.

Esta pesquisa propõe uma métrica *fuzzy* em um algoritmo para aprendizagem de estrutura de redes bayesianas a partir dos dados, aqui denominado MCMC-*Fuzzy*. A Figura 5 apresenta a estrutura dos procedimentos metodológicos realizados nesta pesquisa.

Figura 5 – Procedimentos metodológicos



Fonte: Elaborada pelo autor

As etapas para o desenvolvimento do método iniciaram com a investigação de abordagens utilizadas na tarefa de aprendizagem de estrutura de redes bayesianas. Entre os pesquisados e já descritos em seções anteriores, os algoritmos baseados em busca heurística com funções de avaliações são os mais utilizados.

Assim, esta pesquisa constatou que os algoritmos desse método aplicam métricas de avaliação únicas. Ao utilizar apenas uma métrica, o algoritmo encontra a rede que melhor representa a propriedade que esta métrica avalia.

Uma maneira de investigar o processo de aprendizagem é alterar o processo de avaliação de estruturas, considerando diferentes métricas que avaliam propriedades distintas simultaneamente em uma única função de avaliação. Desta forma, esta pesquisa busca modelar uma métrica *fuzzy* que combine diferentes métricas.

A métrica *fuzzy* irá combinar as diferentes métricas em um único valor. Os conjuntos *fuzzy* são definidos considerando os valores de cada métrica para a rede totalmente conectada e para a rede sem conexões. A função triangular, com três conjuntos *fuzzy*, foi escolhida por ser uma das mais utilizadas. Além disso obteve melhores resultados quando comparada a função gaussiana.

A função de pertinência utilizada para a variável de saída foi a gaussiana. Foram definidos quatro conjuntos *fuzzy* com base nas funções de pertinência das métricas. As regras foram definidas com base nas propriedades das métricas utilizadas.

O motor de inferência utilizado foi o *Mamdani*, sendo este o mais utilizado em sistemas *fuzzy*. Também foram realizados testes comparando *Mamdani* e *Larsen*. O método de agregação aplicado nesta pesquisa foi o de máximo, sendo que ele foi comparado ao operador de soma.

O passo final da métrica *fuzzy* é calcular o valor de saída. Entre os diferentes métodos de desfuzzyficação, o centroide é tem ampla utilização e foi aplicado nesta pesquisa. Outros métodos de desfuzzyficação foram testados, como a média dos máximos, mínimo dos máximos e máximo dos máximos[1].

Após definir a métrica *fuzzy*, foi realizada uma análise para selecionar um algoritmo de busca que iria aplica-la. O método escolhido nesta etapa foi o MCMC, pois consegue lidar com problemas de busca complexos, sendo utilizado em diversos outros algoritmos de pesquisas recentes (BROOKS, 2011)

O método deve ser capaz de encontrar melhores estruturas de redes bayesianas ao combinar diferentes métricas que avaliam diferentes propriedades em uma métrica *fuzzy*.

---

[1] Para mais informações sobre lógica *fuzzy*, veja Russel, Norvig (2009).

A avaliação da acurácia do método foi realizada comparando os resultados encontrados com os de outros algoritmos em bases de dados sintéticas.

A acurácia á analisada comparando a rede padrão ouro com a rede aprendida a partir dos dados. Desta forma, temos as quantidades de arcos aprendidas pelo algoritmo, de arcos corretos, que devem existir na rede, arcos extras, de relações aprendidas porém inexistentes na rede real, e faltantes, que deveriam existir na rede, porém o algoritmo não foi capaz de identificar.

A avaliação também considerou o tempo de execução dos algoritmos e a convergência das cadeias dos métodos MCMC por meio dos métodos estatísticos de Geweke e Heidelberg-Welch.

O método de Geweke foi apresentado em 1992 e utiliza duas partes da cadeia de Markov. As duas partes, normalmente 10% do início e 50% do fim da cadeia, são comparadas a fim de verificar se estas pertencem a mesma distribuição.

O teste calcula a diferença das médias das duas partes dividida pelo erro padrão, considerando que as duas partes são independentes. Valores altos para este teste indicam rejeição (GEWEKE, 1992).

O teste de Heidelberg-Welch foi apresentado em 1983, este método também testa convergência de cadeias de Markov. O método utiliza dois passos, se reprovar no primeiro, o segundo teste não é executado.

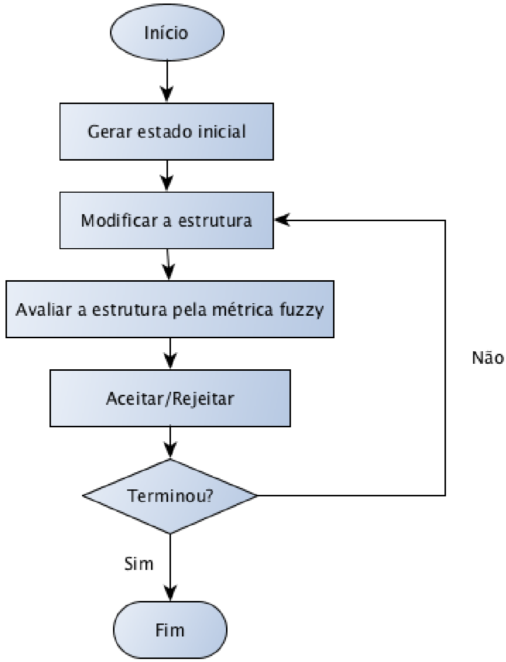
A segunda parte tenta criar um intervalo de confiança a partir da cadeia. Se a divisão entre a largura a meia altura e a média da cadeia for menor que um determinado valor *e*, então a cadeia passa também no segundo teste. O valor de *e* deve ser informado como parâmetro, sendo que seu valor padrão é 0,1. A largura a meia altura é definida como a medida na metade de uma curva (HEIDELBERGER; WELCH, 1983).

A implementação foi realizada utilizando a linguagem de programação R, pois esta linguagem fornece vários métodos para manipulação e analise de dados.

4.2 IMPLEMENTAÇÃO

A implementação do método MCMC-*Fuzzy* foi realizada considerado o fluxograma apresentado na Figura 6.

Figura 6 – Fluxograma do método MCMC-*Fuzzy*



Fonte: Elaborada pelo autor

Dado um conjunto de dados, a primeira etapa do algoritmo gera uma estrutura para a rede bayesiana, sendo este o estado inicial da cadeia de Markov.

A seguir duas variáveis são amostradas aleatoriamente a partir do conjunto de variáveis. A modificação na estrutura é feita gerando um número aleatório a partir de uma distribuição uniforme.

As modificações possíveis são adicionar, remover ou inverter um arco entre as variáveis amostradas. Como uma rede bayesiana não pode conter ciclos – antes de avaliar a estrutura – o algoritmo faz esta validação.

Ao gerar uma amostra de estrutura válida o algoritmo avalia esta estrutura. Geralmente a estrutura é avaliada utilizando uma única métrica, caso do algoritmo EM-MCMC. Algumas métricas favorecem estruturas mais complexas, outras levam em consideração a quantidade de parâmetros. A proposta é combinar métricas conhecidas em uma modelagem *fuzzy* para avaliar as estruturas no método MCMC.

A motivação para utilizar a lógica *fuzzy* está em avaliar simultaneamente a estrutura utilizando diferentes métricas. A modelagem *fuzzy* fornece uma forma flexível de combinar estas métricas.

A modelagem *fuzzy* irá combinar as três métricas descritas, AIC, MDL e BDe. A partir dessas métricas, conjuntos *fuzzy* são definidos. Assim a métrica terá quatro variáveis, as três métricas de entrada e uma variável de saída denominada Qualidade.

Quadro 2 – Conjuntos fuzzy

Variáveis	**Conjuntos *fuzzy***
AIC	Baixa, Moderada e Alta
MDL	Baixa, Moderada e Alta
BDe	Baixa, Moderada e Alta
Qualidade	Ruim, Regular, Boa e Excelente

Fonte: Elaborado pelo autor

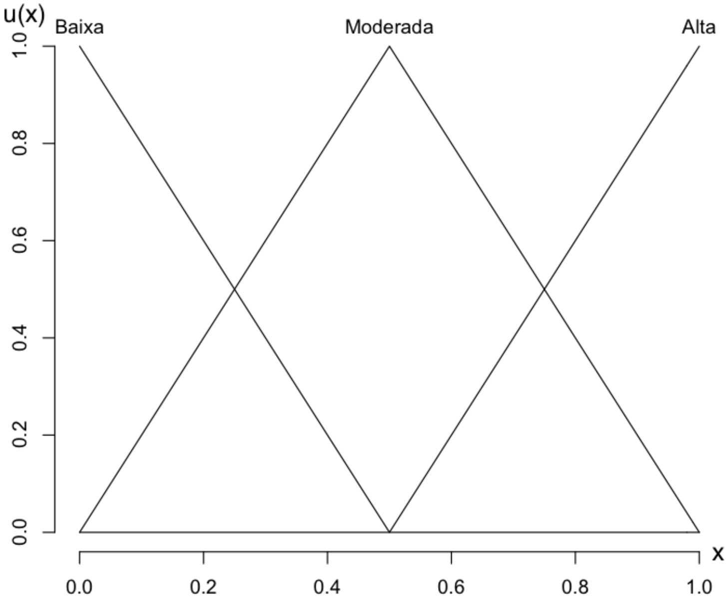
Cada conjunto é definido por uma função de pertinência. A definição das funções de pertinência foi feita utilizando o maior valor entre as métricas considerando uma estrutura de rede totalmente conectada e a mesma estrutura totalmente desconectada. A fim de facilitar a implementação todos os conjuntos foram definidos uniformemente no intervalo [0,1]. A equação 18 apresenta o cálculo da normalização utilizada nos valores das métricas.

$$\mathrm{N} = \frac{\mathrm{v} - \mathrm{v_{min}}}{\mathrm{v_{max}} - \mathrm{v_{min}}} \quad (18)$$

Onde:
v = valor a ser normalizado;
v_{min}= valor mínimo;
v_{max}= valor máximo.

A Figura 7 apresenta a função de pertinência para as métricas. Onde x representa o valor da métrica e $\mu(x)$ a sua pertinência.

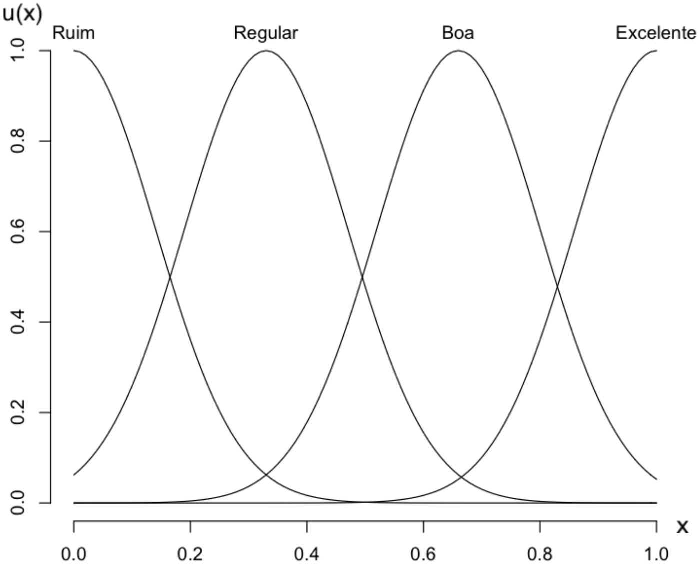
Figura 7 – Funções de pertinências para as métricas



Fonte: Elaborada pelo autor

A função de pertinência para a variável Qualidade é apresentada a Figura 8.

Figura 8 – Função de pertinência para a variável de saída



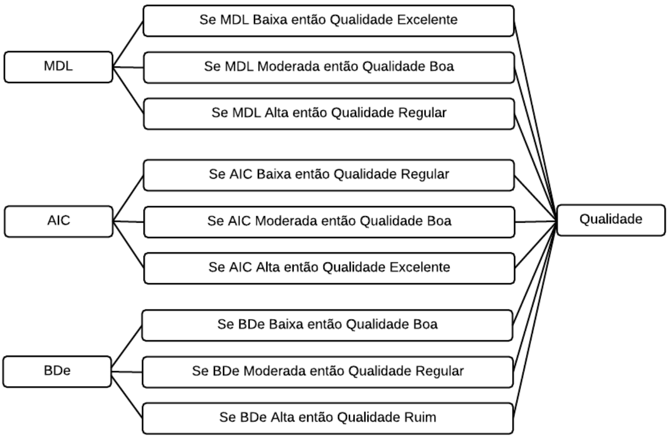
Fonte: Elaborada pela autor

A próxima etapa é definir a base de regras *fuzzy*. As métricas MDL e BDe devem ser minimizadas, a AIC maximizada. A especificação das regras foi feita de forma subjetiva. Assim, a métrica BDe influencia os conjuntos Ruim, Regular e Boa da variável Qualidade. As métricas MDL e AIC os conjuntos Regular, Boa e Excelente. Sendo esta a configuração que obteve melhores resultados.

A

Figura 9 apresenta as regras.

Figura 9 – Regras para a métrica *fuzzy*



Fonte: Elaborada pelo autor

As métricas são agregadas à variável Qualidade por truncamento. A parte final seria a desfuzzyficação da variável de saída Qualidade. O método selecionado foi o centróide.

O algoritmo continua até que a quantidade de iterações seja atingida. O pseudocódigo do algoritmo é apresentado na Figura 10.

Figura 10 – Pseudocódigo do algoritmo

Algoritmo 1 MCMC-fuzzy

```
1: procedure MCMC-FUZZY
2:     rede ← gerarEstadoInicial()
3:     pontuacao ← calcularPontuacaoFuzzy(rede)
4:     melhorPontuacao ← pontuacao
5:     while iteracao < numeroIteracoes do
6:         no1 ← selecionarNo(rede)
7:         no2 ← selecionarNo(rede)
8:         operacao ← selecionarOperacao()
9:         novaRede ← alterarRede(rede, no1, no2, operacao)
10:        novaPontuacao ← calcularPontuacaoFuzzy(rede)
11:        if uniforme() < min(1, novaPontuacao/pontuacao) then
12:            melhorPontuacao ← pontuacao
13:            rede ← novaRede
14:        end if
15:    end while
16:    return G^R
17: end procedure
```

Fonte: Elaborada pelo autor

O método calcularPontuacaoFuzzy() calcula as métricas de pontuação individuais, e por meio da métrica *fuzzy* retorna um valor para a variável Qualidade.

O algoritmo inicia gerando uma estrutura aleatória. A linha 3 calcula a pontuação desta rede, a na linha 4 guarda esta como sendo a melhor. Após estes passos é possível iniciar o processo iterativo.

As linhas 6 e 7 selecionam aleatoriamente dois nós da rede, a linha 8 uma das operações disponíveis, adicionar, remover ou inverter arcos. A operação é realizada na linha 9. A linha 10 calcula a pontuação para a nova amostra de rede. A linha 11 verifica se um número aleatório de uma distribuição uniforme é menor que o mínimo entre 1 e o exponencial da diferença entre as pontuações. Se for então a estrutura é aceita como nova amostra.

O teste apresentado na linha 11 representa a probabilidade de aceitação da cadeia conforme equação 17.

5 RESULTADOS

O método MCMC-*Fuzzy* foi aplicado a duas bases de dados sintéticas. As bases de dados sintéticas foram escolhidas por serem comumente utilizadas para validação de algoritmos de aprendizagem de estrutura de redes bayesianas.

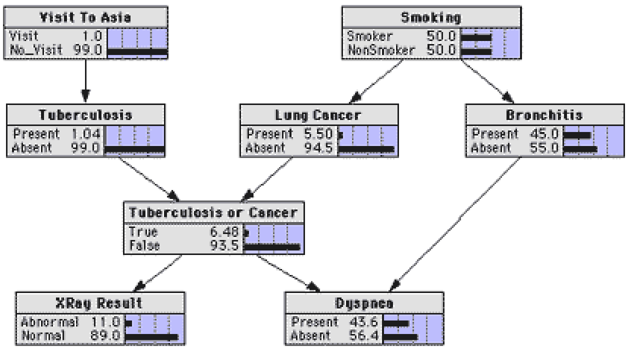
A avaliação para bases de dados sintéticas foi realizada analisando a acurácia, comparando a rede gerada com a rede real, o tempo de execução e a convergência das cadeias geradas pelos métodos MCMC.

Os resultados obtidos pelo método MCMC-*Fuzzy* também foram comparados com os de outros algoritmos, que utilizam diferentes técnicas para lidar com o problema da aprendizagem de estrutura de redes bayesianas a partir dos dados.

5.1 BASES DE DADOS

Uma das bases de dados sintética representa o diagnóstico de pacientes que chegam na emergência de um hospital. A rede é chamada de Ásia e foi apresentada por Lauritzen e Spiegelhalter em 1988. A Figura 11 apresenta a rede Ásia.

Figura 11 – Rede Ásia – Padrão ouro

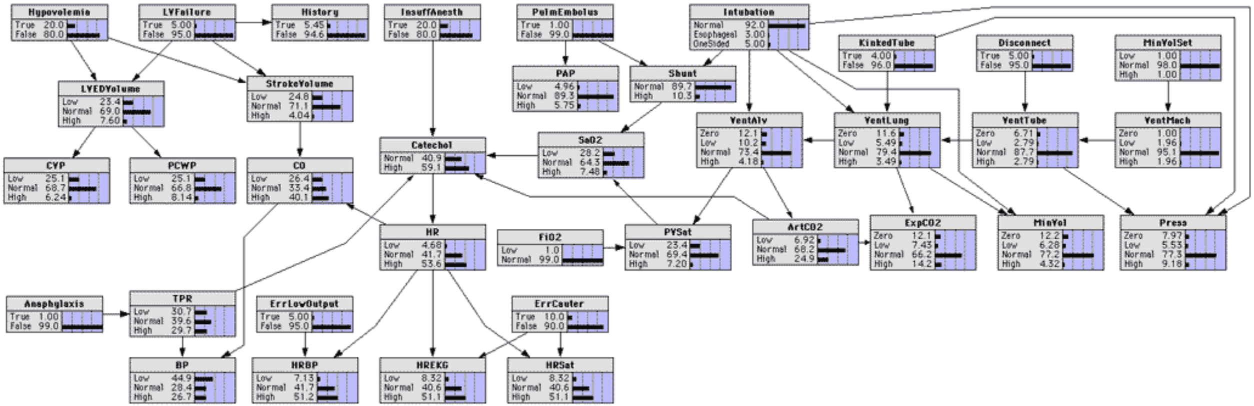


Fonte: Norsys (2014)

A rede Ásia possui oito variáveis e oito conexões. Segundo seus autores, a rede descreve que dispneia ou falta de ar pode ser causada por tuberculose, câncer de pulmão ou bronquite. Uma recente visita a Ásia aumenta as chances de tuberculose. Fumar pode causar câncer de pulmão ou bronquite. O exame de raio X não discrimina câncer de pulmão ou bronquite, nem se existe ou não dispneia.

Outra base muito utilizada em testes de algoritmos de aprendizagem de estruturas é a base Alarme. Esta base foi descrita por Beinlich et al. (1989) e é apresentada na Figura 12.

Figura 12 – RB Alarme – Padrão ouro



Fonte: Norsys (2014)

A rede Alarme foi descrita para apresentar uma mensagem de alarme em um sistema de monitoramento de pacientes. Entre as variáveis, existem oito diagnósticos, dezesseis informações sobre o estado do paciente e treze variáveis intermediárias; com quarenta e seis conexões.

5.2 AVALIAÇÃO

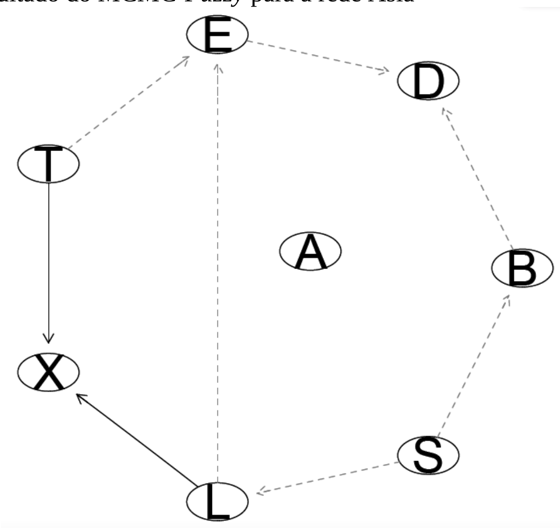
A avaliação para as bases de dados foi realizada primeiramente comparando as estruturas aprendidas pelo método proposto, MCMC-*Fuzzy*, com os algoritmos EM-MCMC, MMHC e K2, comumente utilizados para esta tarefa. Os métodos MCMC tiveram suas cadeias avaliadas graficamente e por métodos estatísticos. O tempo de execução dos algoritmos também foi avaliado.

5.2.1 Comparação com outros algoritmos

A Figura 13 apresenta o resultado encontrado pelo método MCMC-*Fuzzy* para a base de dados Ásia. Os arcos tracejados representam arcos identificados corretamente pelo algoritmo.

As variáveis da rede foram representadas por suas iniciais.

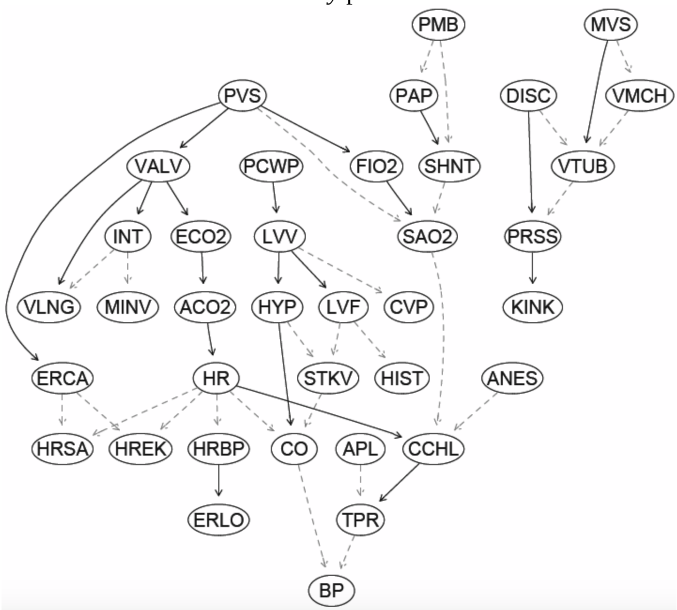
Figura 13 – Resultado do MCMC-*Fuzzy* para a rede Ásia



Fonte: Elaborada pelo autor

A Figura 14 mostra a rede bayesiana identificada pelo método MCMC-*Fuzzy* para a base de dados Alarme.

Figura 14 – Resultado do MCMC-*Fuzzy* para a rede Alarme



Fonte: Elaborada pelo autor

As tabelas 1 e 2 apresentam os resultados encontrados pelos algoritmos K2, MMHC, EM-MCMC e o método MCMC-*Fuzzy* para as bases de dados Ásia e Alarme. Os algoritmos EM-MCMC e MCMC-*Fuzzy* tem como condição de parada o número de iterações, assim, os testes foram realizados com diferentes configurações para este parâmetro. O padrão ouro apresenta a configuração correta da RB para a base de dados. Os algoritmos utilizados para comparação foram escolhidos devido a sua popularidade, eficiência e ampla utilização.

Tabela 1 – Resultados para a rede Ásia

Algoritmos	Total	Arcos corretos	Arcos extras	Arcos faltantes
K2	8	7	1	1
MMHC	6	4	2	4
EM-MCMC	8	4	4	4
MCMC-*Fuzzy*	8	6	2	2
Padrão ouro	8	8	0	0

Fonte: Elaborada pelo autor

Tabela 2 – Resultados para a rede Alarme

Algoritmos	Total	Arcos corretos	Arcos extras	Arcos faltantes
K2	46	45	1	1
MMHC	35	18	17	28
EM-MCMC	36	19	17	27
MCMC-*Fuzzy*	46	26	20	20
Padrão ouro	46	48	0	0

Fonte: Elaborada pelo autor

O algoritmo K2 é muito conhecido por apresentar bons resultados em diversas bases de dados. Isto acontece devido ao parâmetro de entrada que define a ordem das variáveis da rede bayesiana. Definir este parâmetro é uma tarefa complexa, considerando que a aprendizagem é geralmente utilizada quando se tem pouca ou nenhuma informação sobre os dados. Os outros algoritmos constroem a rede bayesiana utilizando apenas os dados, sem necessidade de parâmetros adicionais, o que torna a aprendizagem mais difícil.

O algoritmo MMHC é utilizado em diversas aplicações, principalmente por ser escalável, tendo bons tempos de processamento mesmo quando o número de variáveis é alto. Aplicado às bases de dados Ásia e Alarme o algoritmo identificou 4 arcos corretos para a base Ásia e 18 para a base Alarme.

Os resultados do MMHC mostram certa equivalência com os resultados encontrados pelo algoritmo EM-MCMC, identificando quase as mesmas quantidades de arcos corretos, extras e faltantes em ambas as bases.

O algoritmo MCMC-*Fuzzy* obteve bons resultados para as duas bases de dados. Aplicado a base de dados Ásia, o algoritmo identificou 6 arcos corretos, tendo resultado compatível com o algoritmo K2, sendo que o método proposto não necessita de parâmetros adicionais.

Para a base de dados Alarme o algoritmo MCMC-*Fuzzy* obteve a maior quantidade de arcos corretos e a menor de arcos faltantes, porém seus resultados ainda foram inferiores ao do algoritmo K2. Também é possível perceber que o algoritmo MCMC-*Fuzzy* identificou mais arcos extras em comparação com os outros métodos. Desta forma, obteve uma estrutura de rede mais complexa, o que pode comprometer a estimação dos parâmetros probabilísticos da rede.

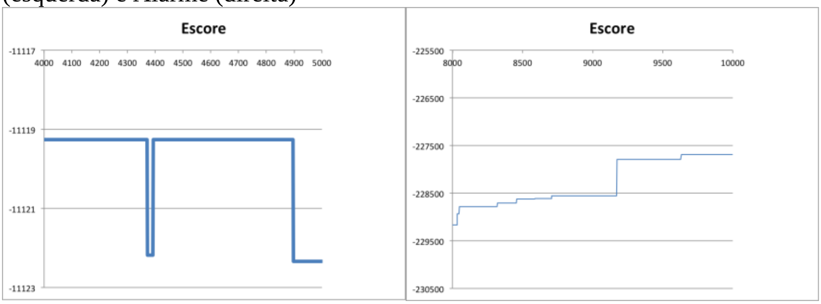
Uma possível verificação seria comparar as estruturas geradas pelos diferentes métodos por meio da acurácia de cada estrutura. Para isto, seria necessário estimar os parâmetros probabilísticos da rede, o que foge ao escopo desta pesquisa.

5.2.2 Convergência da cadeia

Cadeias de Markov podem ser avaliadas graficamente ou por métodos estatísticos. Os dados utilizados para analise da convergência das cadeias são as saídas da métrica de avaliação utilizada nos algoritmos EM-MCMC e MCMC-*Fuzzy*.

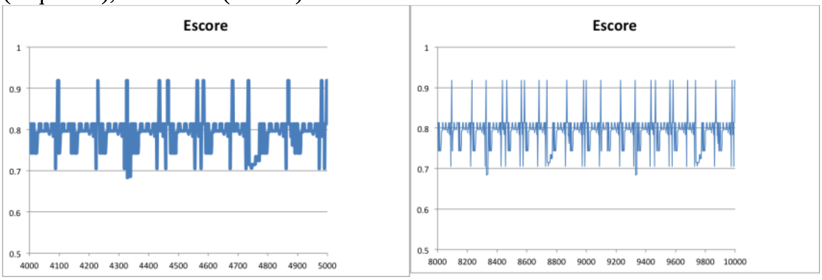
Os gráficos das Figuras 13 e 14 apresentam a convergência da cadeia a seu estado estacionário. Uma cadeia alcança seu estado estacionário quando alterações geram pouca variação em sua saída.

Figura 15 – Convergência do algoritmo EM-MCMC para as redes Ásia (esquerda) e Alarme (direita)



Fonte: Elaborada pelo autor

Figura 16 – Convergência do método MCMC-*Fuzzy* para as redes Ásia (esquerda), e Alarme (direita)



Fonte: Elaborada pelo autor

As cadeias também foram avaliadas utilizando métodos estatísticos. Os métodos escolhidos foram o de Geweke e Heidelberg-Welch. O teste de Geweke rejeita valores altos, o de Heidelberg apresenta os valores do primeiro passo, da média e da largura a meia altura.

Os testes de Geweke e Heidelberg-Welch foram executados por meio da biblioteca CODA, implementada por Plummer et al. (2006). As tabelas 3 e 4 apresentam os resultados destes testes.

Tabela 3 – Avaliação da cadeia para o algoritmo EM-MCMC

Bases	Geweke	Heidelberg-Welch
Ásia	0,8161	0,7/-11120/0,737
Alarme	-4,655	0,0764/-227999/560

Fonte: Elaborada pelo autor

Tabela 4 – Avaliação da cadeia para o método MCMC-*Fuzzy*

Bases	Geweke	Heidelberg-Welch
Ásia	0,04826	0,997/0,795/0,00707
Alarme	-0,0002886	0,998/0,795/0,00477

Fonte: Elaborada pelo autor

Os valores encontrados para o teste de Geweke foram baixos, indicando aceitação da cadeia. O teste de Heidelberg-Welch apresenta os valores para o teste inicial, média, e largura a meia altura, respectivamente. O valor do teste inicial tenta verificar se a cadeia chegou em seu estado estacionário, apenas se a cadeia passar neste teste a segunda etapa é executada. A segunda divide a largura a meia altura e a média, se o resultado for menor que um parâmetro *e*, sendo que o utilizado foi 0,1, então a cadeia passa na segunda etapa do teste.

Segundo os testes realizados, os dois algoritmos chegaram ao estado estacionário de suas cadeias de Markov.

5.2.3 Tempo de execução

Estes testes foram realizados em um computador com processor Intel Core i7, com 8GB de memória RAM rodando o sistema operacional MAC OS X 10.9.

Os testes de tempo de execução dos algoritmos foram realizados com as mesmas bases utilizadas nos testes das redes geradas. Os algoritmos K2 e MMHC possuem um critério de parada, enquanto os algoritmos EM-MCMC e o método MCMC-*Fuzzy* utilizam um método MCMC e por isso dependem da convergência da cadeia. A convergência é influenciada pelo seu estado inicial, que é aleatório. Desta forma, os testes consideraram diferentes números de iterações para esses algoritmos, sendo apresentados na tabela os tempos com os melhores resultados. Os algoritmos foram rodados por 5000 iterações para a base de dados Ásia e 10000 iterações para a base de dados Alarme.

Tabela 5 – Comparação do tempo de execução em segundos

Algoritmos	Ásia	Alarme
K2	1,5456	3,6432
MMHC	0,0385	1,2232
EM-MCMC	2,9568	15,1015
MCMC-*Fuzzy*	5,0924	27,794

Fonte: Elaborada pelo autor

O algoritmo MMHC é conhecido por obter ótimos tempos de processamento, sendo muito utilizado em grandes bases de dados. O método MCMC-*Fuzzy* obteve o maior tempo de processamento, devido a métrica *fuzzy*.

Analisando os tempos de processamento, é possível perceber que a melhora na aprendizagem da estrutura teve um custo, mesmo que aceitável.

É possível perceber que além do número de iterações para os métodos MCMC, o tamanho da base de dados influencia diretamente em seu tempo de execução.

O algoritmo K2 obteve os melhores resultados nos experimentos realizados, porém sua execução depende de conhecimento prévio sobre os dados, já que a ordenação das variáveis deve ser informada como parâmetro de entrada.

O método MCMC-*Fuzzy* se mostrou eficiente obtendo os melhores resultados entre os algoritmos que aprendem a rede bayesiana apenas utilizam dados.

6 CONCLUSÃO

Esta pesquisa investigou o desenvolvimento de uma métrica *fuzzy* aplicada a um método para aprendizagem de estruturas de redes bayesianas de forma automatizada.

O algoritmo MCMC-*Fuzzy* combina diferentes funções de avaliação de redes bayesianas em uma métrica *fuzzy*, por meio de um método MCMC. A combinação de diferentes funções de avaliação resultou em uma métrica que considera diferentes propriedades de redes bayesianas simultaneamente. A métrica modelada adicionou pouca complexidade ao método, utilizando poucas regras. Além se independente do método de busca, podendo ser aplicada em qualquer outro algoritmo que utilize métricas individuais.

A avaliação foi realizada usando duas bases de dados, escolhidas por serem comumente utilizadas em testes com algoritmos de aprendizagem de redes bayesianas.

O método MCMC-*Fuzzy* obteve bons resultados, sendo que apenas o algoritmo K2 o superou. O algoritmo K2 possui como parâmetro de entrada a ordem das variáveis, o que facilita a aprendizagem, sendo que os outros algoritmos não necessitam de parâmetros adicionais. Informar a ordem das variáveis requer conhecimento prévio sobre os dados, sendo esta a principal desvantagem do algoritmo K2.

Considerando os algoritmos que não necessitam de parâmetros adicionais o método MCMC-*Fuzzy* obteve os melhores resultados para a base de dados Ásia. Para a base de dados Alarme o método identificou a maior quantidade de arcos corretos. Porém ainda houve muitos arcos faltantes, que deveriam ter sido identificados, e arcos extras que não deveriam existir na estrutura da rede. O algoritmo MMHC identificou menos arcos corretos, mas também encontrou menos arcos extras.

Os resultados mostram que o algoritmo MCMC-*Fuzzy* obtém bons resultados para uma rede simples, como a Ásia. Porém para uma rede mais complexa, como a base Alarme, o método identificou muitos arcos extras em comparação com os outros métodos, obtendo uma estrutura de rede mais complexa.

A fim de identificar a convergência das cadeias geradas pelo método MCMC, dois métodos estatísticos foram aplicados a estas cadeias, comprovando que as cadeias convergiram a seus estados estacionários.

Em relação ao tempo de execução dos algoritmos, o algoritmo MMHC obteve os melhores resultados e o método MCMC-*Fuzzy* obteve tempo de processamento compatível com outros métodos, mesmo com os piores resultados.

Uma dificuldade no desenvolvimento do método MCMC-*Fuzzy* foi a definição do conjunto de regras e dos parâmetros das funções de pertinência.

Como sugestão para trabalhos futuros, indica-se:

- Avaliar a acurácia do método MCMC-*Fuzzy* em relação a aprendizagem de parâmetros de RB;
- Investigar outras funções de avaliação na modelagem *fuzzy* da métrica;
- Avaliar a utilização de números *fuzzy* para combinar as métricas, substituindo a métrica *fuzzy* com regras;
- Investigar a métrica *fuzzy* aplicada a outros algoritmos de aprendizagem de estrutura pelo método de busca heurística, como por exemplo, algoritmos genéticos.

REFERÊNCIAS

AKAIKE, H. **A New Look at the Statistical Model Identification**. Automatic Control, IEEE Transactions p. 716-723, 1974.

ALIFERIS, C. F.; STATNIKOV, A.; TSAMARDINOS, I.; MANI, S.; KOUTSOUKOS, X. D. **Local causal and markov blanket induction for causal discovery and feature selection for classification part i: Algorithms and empirical evaluation**. In: The Journal of Machine Learning Research, v. 11, p. 171-234, 2010.

ALONSO-BARBA, J. I., DE LA OSSA, L., GÁMEZ, J. A., PUERTA, J. M. **Scaling up the greedy equivalence search algorithm by constraining the search space of equivalence classes**. In: Symbolic and Quantitative Approaches to Reasoning with Uncertainty, p. 194-205, Springer Berlin Heidelberg, 2011.

BEINLICH I.; SUERMONDT, HJ; CHAVEZ, RM; COOPER, GF. **The ALARM Monitoring System: A Case Study with Two Probabilistic Inference Techniques for Belief Networks**. Springer Berlin Heidelberg, 1989.

BOUCKAERT, Remco R. **Probabilistic Network Construction Using The Minimum Description Length Principle**. Technical Report, Utrecht University, 1994.

BROOKS, Steve; GELMAN, Andrew; JONES, Galin L.; MENG, Xiao-Li. **Handbook of Markov Chain Monte Carlo**. New York: CRC Press, 2011.

CASTILLO, Enrique. **Expert systems and probabilistic network models**. Springer Verlag , 1997.

CHENG, J.; GREINER, R.; KELLY, J.; BELL, D.; LIU, W. Learning Bayesian networks from data: An information-theory based approach, In: **The Artificial Intelligence Journal**, v. 137, p. 43-90, 2002.

CHENG, Jie; GREINER, Russel. **Learning Bayesian Belief Network Classifiers: Algorithms and System**. Advances in Artificial Intelligence: Springer Berlin Heidelberg, 2001.

CHICKERING, David Maxwell. Optimal structure identification with greedy search. In: **The Journal of Machine Learning Research**, p.507-554, 2003.

CHICKERING, David Maxwell. **Learning Bayesian networks is NP-complete**. In Learning from data. Springer New York. p. 121-130. 1996.

COSTA, Felipe Schneider. **Aprendizagem Estrutural de Redes Bayesianas pelo Método de Monte Carlo e Cadeias de Markov**. Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Santa Catarina - Dissertação, 2013.

CROTTI JUNIOR, Ademar; WILGES, Beatriz; NASSAR, Silvia Modesto. **Modelagem bayesiana da aprendizagem de estudantes em um AVA**. Simpósio Brasileiro de Informática na Educação (SBIE), 2014, Dourados - MS. Anais do XXV SBIE, 2014.

DALY, Rónán; QIANG Shen; STUART, Aitken. Learning Bayesian networks: approaches and issues. In: **The Knowledge Engineering** Review: 99-157, 2011.

FAULKNER, Eli. K2GA: **Heuristically Guided Evolution of Bayesian Network Structures from Data**. CIDM. 2007.

FREITAS FILHO, Paulo José. **Introdução à Modelagem e Simulação de Sistemas com Aplicações Arena**. Florianópolis: Visual Books, 2008.

FRIEDMAN, Nir; KOLLER, Daphne. **Being Bayesian about network structure. A Bayesian approach to structure discovery in Bayesian networks**. In: Machine learning, v. 50, n. 1-2, p. 95-125, 2003.

GÁMEZ, José A.; JUAN L. Mateo; PUERTA, José M. Learning **Bayesian networks by hill climbing: efficient methods based on progressive restriction of the neighborhood**. In: Data Mining and Knowledge Discovery v. 22, n. 1-2, p. 106-148, 2011.

GEWEKE, J. **Evaluating the Accuracy of Sampling-Based Approaches to Calculating Posterior Moments**. In: Bernado JM, et al., editors. Bayesian Statistics 4. 1992.

GRZEGORCZYK, Marco; HUSMEIER, Dirk. **Improving the structure MCMC sampler for Bayesian networks by introducing a new edge reversal move**. Machine Learning v. 71, n. 2-3: p. 265-305, 2008.

GUO, Peng; LI Naixiang. **An EM-MCMC algorithm for Bayesian structure learning**. In: Computer Science and Information Technology, 2009. ICCSIT 2009. 2nd IEEE International Conference on. IEEE, 2009.

HECKERMAN, David; GEIGER, Dan; CHICKERING, David M. **Learning Bayesian networks: The combination of knowledge and statistical data**. In: Machine learning 20.3. p. 197-243, 1995.

HEIDELBERGER, P.; WELCH. P.D. **Simulation Run Length Control in the Presence of an Initial Transient**. In: Operations Research v. 31, p. 1109-1144. 1983.

HÜLLERMEIER, Eyke. ***Fuzzy* sets in machine learning and data mining**. In: Applied Soft Computing**,** v.11, n. 2. p. 1493-1505, 2011.

KO, Song, KIM, Dae-Won. **An efficient node ordering method using the conditional frequency for the K2 algorithm**. In: Pattern Recognition Letters, 2014.

LARRAÑAGA, P.; KARSHENAS, H.; BIELZA, C.; SANTANA, R. **A review on evolutionary algorithms in Bayesian network learning and inference tasks**. In: Information Sciences, 233, 109-125, 2013.

LAURITZEN, Steffen L. SPIEGELHALTER, David J.. **Local computations with probabilities on graphical structures and their application to expert systems**. In: Journal Royal Statistics Society B. 157-194, 1998.

MARGARITIS, D., THRUN, S. **Bayesian network induction via local neighborhoods**. In: Adv. Neural Inf. Process. Systems v. 12, p. 505–511, 1999.

MARSLAND, Stephen. **Machine Learning: an algorithmic perspective**. New York: Chapman & Hall, 2009.

MASEGOSA, Andrés R., MORAL, Serafín. **New skeleton-based approaches for Bayesian structure learning of Bayesian networks**. Applied Soft Computing13.2: 1110-1120, 2013.

MORALES, Manuel Martinez, et al. **A method based on genetic algorithms and *fuzzy* logic to induce Bayesian networks**. Computer Science, 2004. ENC 2004. Proceedings of the Fifth Mexican International Conference In. IEEE, 2004.

MURPHY, Kevin P. **Machine learning: a probabilistic perspective**. MIT Press, 2012.

NEAPOLITAN, R. E. **Learning bayesian networks**. Pearson Prentice Hall Upper Saddle River, 2004.

NIELSEN, Jens D.; KOCKA, Tomás; PENA, Jose M. **On local optima in learning Bayesian networks**. Proceedings of the Nineteenth conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann Publishers Inc., 2002.

NIINIMAKI, Teppo; PARVIAINEN, Pekka; KOIVISTO, Mikko. **Partial order MCMC for structure discovery**. In: Bayesian networks, 2012.

NORSYS, **Netica Bayesian Networks Software from Norsys**. Disponível em: <http://www.norsys.com>. Acesso em: 9 set 2014.

PEARL, Judea. **Probabilistic Reasoning in Inteligent Systems**. New York: Morgan Kaufmann, 1988.

PELLET, J.P.; ELISSEEFF, A. **Using Markov blankets for causal structure learning**. In: Machine Learning Research. Res. 9, 1295–1342, 2008.

PLUMMER, Martyn; BEST, Nicky; COWLES, Kate; VINES, Karen Vines. **CODA: Convergence Diagnosis and Output Analysis for MCMC**. In: R News, v.6, p. 7-11, 2006.

RUSSELL, Stuart Jonathan; NORVIG, Peter. **Artificial intelligence**: a modern approach. 3. ed. London: Prentice-Hall, 2009.

SCHWAIGHOFER, A., DEJORI, M., TRESP, V., STETTER, M. **Structure learning with nonparametric decomposable models**. In: Artificial Neural Networks–ICANN. Springer Berlin Heidelberg, 119-128, 2007.

SESTATNET. **Ensino-aprendizagem de estatística na web**. Disponível em: <http://www.sestatnet.ufsc.br/>. Acesso em: 20 maio 2014.

SIVANANDAM, S. N.; SUMATHI S.; DEEPA, S. N.. **Introduction to *fuzzy* logic using MATLAB**. Vol. 1. Berlin: Springer, 2007.

SMITH, Jim Q. **Bayesian decision analysis: principles and practice**. Cambridge University Press, 2010.

TSAMARDINOS, Ioannis; BROWN, Laura E.; ALIFERIS, Constantin F. **The max-min hill-climbing Bayesian network structure learning algorithm**. In: Machine learning, v. 65, n.1, p. 31-78, 2006.

YAN, Lisa Jing; CERCONE, Nick. **Bayesian network modeling for evolutionary genetic structures**. In: Computers & Mathematics with Applications**,** v. 59, n. 8, p. 2541-2551, 2010.

YUAN, Changhe; MALONE, Brandon. **Learning Optimal Bayesian Networks: A Shortest Path Perspective**. In: Journal of Artificial Intelligence Research, v. 48, p. 23-65, 2013.

ZHANG, Yinghua; ZHANG, Wensheng; XIE, Yuan. **Improved heuristic equivalent search algorithm based on Maximal Information Coefficient for Bayesian Network Structure Learning**. In: Neurocomputing**,** 117, p.186-195, 2013.